RECEIVED
CENTRAL FAX CENTER
DEC 3 0 2004

To: Joshua D Campbell

Fax #: +1 (703) 872-9306

From: George L. Yang

Fax #: 410-581-0969

Date/Time: 12/30/2004 2:27:36 PM

Subject: Amendment Draft

Pages: 22

Attn: Joshua D Campbell
U.S. Department of Commerce
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450
Fax: 703-872-9306

Subject: Proposed Amendment

From: George L . Yang
4628 Kings Mill Way,
Owings Mills, MD 21117
Tel/Fax: 410-581-0969
Email: george_yang@yahoo.com
Date: 2004/12/30

Dear Joshua,

I proposed the amendment draft. Since I have been an engineer for more than 20 years but I have no experience in writing claims, even though I have spent a lot of time to read references and related documents, I may again make some mistakes. I sincerely and respectively request you to help me to fix any problems associated with the claims. I will call you in about 10 days to listen your opinions.

Have a wonderful new year!

Very respectfully,

George L. Yang

RECEIVED
CENTRAL FAX CENTER
DEC 3 0 2004

In the United States Patent and Trademark Office

Appn. Number: 09/994,548
Appn. Filed: 2001 November 27
Applicant: George L. Yang
Title: METHOD AND SYSTEM FOR A GENERAL COMPUTING SOFTWARE SYSTEM
Examiner/AU: Joshua D Campbell/2179

Faxed: 2004 December 30, Thursday

Amendment A Draft

Organization TC2100 Bldg/Room PK2
U.S. Department of Commerce
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

The applicant would like to thank the Examiner for his listing related materials and pointing out the deficiencies in the original claims. The applicant also greatly appreciates the suggestions on how to rewrite claims that the Examiner provided during the phone conversation on December 20, 2004.

In response to the Office Action mailed 2004 December 3, please amend the above application as follows:

Claims: Cancel all the claims of record and substitute new claims 21 to 40 as follows.

21. A general computing software system for helping a user to extract information from an information source, generate programming source code, and reconstruct information, said information source comprising text sections, figure sections, and formula sections, said system comprising:

means for setting up environment for said user to specify default binding rules, default programming language, default working path, default parameters, and default connotations of mathematics symbols;

means for selecting portions from said information source, said portions comprising formulas and figures;

means for recognizing symbols, texts, formulas, and figures on said portions;

means for selecting a block, said block containing an element selected from a group consisting of a formula and a figure, said element having a plurality of items, and each of said items being a component selected from a group consisting of a symbol, a parameter, a function, an operator, a label, and a curve;

means for creating identifications to distinguish among said items;

means for associating properties to said block and said items to specify desired relations among said items and desired ways of handling said block and said items;

means for modifying said properties to change said desired relations among said items and said desired ways of handling said block and said items;

means for defining new properties and means for assigning said new properties to said block and said items to provide specific requirements on said block and said items;

means for processing curve to separate curves, extract information from said curves, and generate information for recreating said curves;

means for processing formula to link functions, identify relations among components of a formula, create a reconstructed formula, and display said reconstructed formula;

means for displaying said block and said items;

means for displaying properties associated with said block and said items;

means for adjusting properties to be assigned to said block and said items by using information obtained from handling previous blocks;

means for generating said programming source code;

means for compiling, linking, and running said programming source code and displaying results from running said programming source code; and

means for updating database.

22. The system according to claim **21**, further comprising means for generating model file and means for modifying said model file, wherein said model file is a file that contains structure information of said programming source code.

23. The system according to claim **21**, further comprising:

means for associating default properties to said items and said block to specify default relations among said items and default ways of handling said items and said block; and

means for modifying said default properties associated with said items and said block to change said default relations and said default ways of handling said items and said block.

24. The system according to claim **21**, further comprising means for adjusting and assigning properties to said previous blocks by making use of information accumulated for handling said block and said items that are under processing, said properties comprising any things associated with said items and needed for extracting information, said items comprising numbers, strings, mathematic symbols, parameters, constants, functions, equations, formulas, figures, and various components of a figure, whereby one property can bind many items, one item can bind many properties, and one property can further bind other properties.

25. The system according to claim **21**, further comprising means for displaying and hiding selected aspects of said block and said items, wherein said aspects comprise properties and their associated information.

26. The system according to claim **21**, wherein said means for processing curve comprises:

means for separating curves into a group of separated curves;

means for selecting a curve from said group of separated curves;

means for assigning properties to said curve to specify how to extract information from said curve and how to create a regenerated curve;

means for extracting parameters associated with said curve;

means for binding said parameters to said regenerated curve;

means for representing said curve by a plurality of ordered pairs; and

means for reconstructing said curve.

27. The system according to claim **26**, wherein said means for reconstructing said curve comprises:

means for identifying if said curve is a continuous curve or a discrete curve;

means for identifying accurate marks on said curve;

means for specifying precision; and

means for specifying interpolation method and finding coefficients associated with said interpolation method from said plurality of ordered pairs.

28. The system according to claim **21**, said system comprising many commonly used functions in its libraries and said block comprising a formula, wherein said means for processing formula comprises:

means for assigning properties to said block and said items;

means for identifying mathematic symbols and functions in said block;

means for identifying relations among said items;

means for displaying said relations;

means for identifying any undefined functions;

means for searching from said libraries for library functions related to said undefined functions;

means for associating an undefined function with a corresponding library function and means for associating arguments of said undefined function with corresponding arguments of said corresponding library function;

means for changing parameters associated with said corresponding library function;

means for creating a regenerated formula;

means for displaying said regenerated formula; and

means for modifying properties of said formula, deleting properties of said formula, defining new properties, and adding said new properties to said formula.

29. The system according to claim **28**, wherein said means for identifying relations among said items comprises means for generating tokens by a scanner and means for creating symbol tables by a parser.

30. The system according to claim **28**, said formula comprising a function and said function having a plurality of variables and a plurality of parameters, further comprising means for assigning as many properties to said function, said variables, and said parameters as needed directly and indirectly, simultaneously and gradually, recursively and non-recursively.

31 The system according to claim 21, wherein said means for displaying properties further comprises:

means for displaying simultaneously all properties related to an item;

mean for displaying sequentially all properties related to an item;

means for displaying a particular property and all items related to said particular property;

means for displaying property's property of an item; and

means for displaying properties in any combination of above means.

32. The system according to claim 29, wherein said means for generating tokens by a scanner, each of said token being an inseparable item associated with certain properties, said scanner being a two dimension processing in nature, whereby said scanner uses not only literal information from an item but also properties associated with said item as well as information obtained from handling other items and previous blocks.

33. The system according to claim 29, wherein said means for generating symbol tables by a parser, said symbol tables being instances of data structures to describe relations among all items of a formula, said parser being a two dimension processing in nature, whereby said parser uses a grammar based on various mathematic structures and properties associated with said items, collects information about said items, and requires specific information from an intelligent source, and inserts extra items into said symbol tables to describe said formula effectively.

34. The system according to claim 21, wherein said means for recognizing symbols, texts, formulas, and figures on said portions is arranged to apply pattern recognition techniques on said portions to identify said symbols, said texts, said formulas, and said figures, wherein said means for updating database further comprises means for selecting and saving model files and programming source code, means for converting data from one format into different formats and saving data in a desired format, means for creating help files, and means for maintaining database; and wherein said means for creating identifications creates new symbols from said symbols, new texts from said texts, new formulas from said formulas, and new figure

from said figures with each of said new symbols, each of said new texts, each of said new formulas, and each of said new figures having a corresponding identification.

35. A computing software system having common used functions in its libraries for a user to regenerate and verify contents in a document, said document comprising a plurality of formula blocks, said system comprising means for identifying said formula blocks, means for processing formula block, means for generating source code, means for compiling, linking and running said source code, and means for displaying results, wherein said means for processing formula block carried out for each of said formula blocks comprises:

means for assigning properties to a formula block selected from said formula blocks and items of said formula block;

means for identifying and selecting an undefined function in said formula block;

means for searching for a corresponding library function in said libraries, and associating said undefined function with said corresponding library function;

means for searching for a previously defined function and associating said undefined function with said previously defined function;

means for identifying relations among said items to create a symbol table for said items;

means for reconstructing said each block to create a corresponding regenerated block; and

means for displaying said regenerated block,

whereby functions in said formula blocks will be linked to corresponding functions in said libraries and corresponding functions defined previously,

whereby means for compiling, linking and running said source code will compile, link, and run said source code to regenerate said results,

whereby means for displaying results will display said results in proper forms, and

whereby said user can compare said results with said contents to verify said contents on said document.

36. The system according to claim **35**, wherein said means for processing formula block further comprises:

means for modifying said properties to give descriptions more specific to said each block and said items; and

means for defining new properties to specify further requirements on said each block and said items.

37. The system according to claim 35, wherein said means for processing formula block further comprises:

means for associating arguments of said undefined function with corresponding arguments of said corresponding library function;

means for specifying parameters associated with said corresponding library function; and

means for displaying properties associated with said regenerated block.

38. A computing software system having common used functions in its libraries for a user to regenerate and verify contents on a document, said document comprising a plurality of curve blocks, each of said curve blocks containing a plurality of curves, said system comprising means for identifying said curve blocks, means for processing curve block, means for generating source code, means for compiling, linking and running said source code, and means for displaying results, wherein said means for processing curve block carried out for each of said curve blocks comprises:

means for assigning properties to a curve block selected from said curve blocks and items of said curve block;

means for separating each curve from a group of curve in said curve block to generate a group of separated curves;

means for selecting one curve from a group of separated curves and specifying properties to said one curve;

means for identifying parameters associated with said one curve and binding said parameters with said one curve;

means for extracting information from said curve; and

means for representing said curve.

39. The system according to claim **38**, wherein said means for extracting information from said curve further comprises:

means for identifying if said curve is a continuous curve or a discrete curve;

means for identifying if said curve has accurate marks on it and means for extracting information from these marks;

means for converting continuous curve into discrete one;

means for specifying an interpolation method; and

means for finding coefficients associated with said interpolation method.

40. The system according to claim **38**, wherein said means for selecting one curve from a group of separated curves further comprises:

means for selecting a curve based on different curve having a different color;

means for selecting a curve based on different curve having a different line pattern;

means for selecting a curve based on different curve having a different line weight; and

means for selecting a curve based on different curve having a different label.

REMARKS -- General

Applicant Has Written Claims 35-40 for Reflecting the Novel Features
According to the suggestions given by the Examiner, the applicant has written claims 35 to 37 to emphasize the novel features to handle equations and formulas based on Fig. 5 in application and the applicant has written claims 38 to 40 to address the innovative features to handle curves in a figure based on Fig. 4 in application.

The Rejection of The Claims 1-2 Under § 103
The applicant has dropped the independent claim 1 and the dependent claim 2.

The Rejection of Claim 13 Under § 102 Is Complied
The applicant has dropped the independent claim 13.

The Rejection of The Claims 7-11 Under § 112 for Being Indefinite
The claim 7 was rejected because of reciting the limitation "pinpointing the problem" and claims 8-11 were rejected because of their base claims.
The applicant has rewritten theses claims, narrowed greatly the base claims, and has fixed the problem of being indefinite.

The Rejections of Claim 3-12 Under § 103

Claims 3-12 were rejected under USC 103(a) as being unpatentable over Pagallo (US Patent Number 5,627,914, issued on May 6, 1997) in view of Koza (UA Patent Number 5,136,686, issued on August 4, 1992). All these rejections are based "obvious" to combining Pagallo's patent and Koza's patent. Claims 3-12 have been rewritten as new claims 21-34 to narrow the claims greatly and define patentably over these references, and any combination thereof. Applicant requests reconsideration of this rejection, as now possibly applicable to claims 21-34, for the following reasons:

(1) There is no justification, in Pagallo or Koza, or any other prior art separate from applicant's disclosure, which suggests that these references can be combined, much less be combined in the manner proposed. Since Koza's patent was earlier than Pagallo's patent., Koza did not refer anything about Pagallo's patent.. Even though Pagallo's patent was much late than Koza's patent, Pagallo also did not discuss anything about Koza's patent and even did not mention Koza's patent at all.

(2) One skilled in the art would have no reason to make such a combination.

(3) Even if Pagallo's patent and Koza's patent could be combined in the manner proposed, the proposed combination would not show all the novel features of claim 21.

(4) These novel features of claim 21 produce new and unexpected results and hence are unobvious and patentable over these references.

The References And Differences Of The Present Invention Thereover

Prior to discussing the claims and the above four points, applicant will first discuss the references and the general novelty of the present invention and its unobviousness over the references.

Koza wrote in the abstract that

> "The present invention is a non-linear generic algorithm for problem solving. The interactive process of the present invention operates on a population of problem solving entities." (underline added for emphasis)

Koza showed how to solve math problem by constructing a very large, non-linear, and non-continuous S-expression populations, finding the best-fit S-expression, and then parsing the best-fit S-expression. On column 33 line 30 to column 34 line 24, Koza wrote:

> "FIG. 9 shows a pair of linear equations having two variables x1 and x2. ...
> The universe from which proffered solutions for the first variable x1 consists of any valid LISP S-expression (with any level of embedding of functions) constructed from the useful input atomic arguments (A11, A21, B1, B2), the extraneous input atomic arguments (A12 and A22), the useful functions of multiplication and subtraction (* and -), and the extraneous function of addition (+). ...

> The search space in which the solution to this problem lies is thus a very large non-linear, non-continuous space of rooted trees whose points are labeled with various valid LISP S-expressions ...
>
> Solving these problems starts by generating a population of individual S-expressions using the functions *, -, and + and the atomic arguments A11, A12, A21 A22, B1 and B2. ...
>
> Once the population of individuals is generated, each of the S-expressions in the population is executed (i.e. evaluated in LISP) to produce a result. ...
>
> The cumulative distance provides a natural measure of value (fitness) of a particular individual S-expression in this environment. If the sum of these distances for particular individual S-expression is zero, the S-expression has the best value (best fitness) in this environment ..." (underlines added for emphasis)

Here Koza first assumed the formula for solving a pair of linear equations was unknown, then Koza built a set of possible S-expression as well as associated rule to measure the fitness and further designed a procedure to find the best-fit S-expression, which later could be translated into program in languages such as FORTRAN COBOL etc

Koza showed one curve on each of Fig. 12, Fig. 13, and Fig. 14 on his patent. Koza wrote

> "Neither the second curve nor the third curve is a good fit to the first curve. ...
> The desired solution to this problem of finding unknown function in symbolic form can be viewed as a search for a function from a hyperspace of functions that can be composed from a set of candidate functions. ...
> In other runs, the symbolic regression was successfully performed on additional target expressions such as ... using function sets containing SIN, COS ...." (underlines added for emphasis)

Again Koza constructed a set of possible S-expression as well as associated rule to measure the fitness, and designed a procedure to find the best fit S-expression, which later could be translated into program in languages such as FORTRAN COBOL etc.

Applicant takes exactly the opposite ways to deal with formulas and equations and create source code.

First, the purpose of applicant's invention is to verify the results appeared on a document and to let user to change parameters and functions to see how the results would change so that user can understand better through the process and save a lot of time without really writing all the codes.

Second, applicant assumes that all functions are available in libraries, have been defined earlier, or will be defined later, or user knows the definitions of any undefined functions

without playing the game of all possible combinations on available constants, symbols, operators, and functions and then finding the one with "best fitness".

Third, applicant lets the user as the ultimate authority to judge if a method used is the best or not by reconstructing an equation or formula and displaying the reconstructed equation or formula as well as associated properties without trying to define the "best fitness". In many situations even in the very simple example of a pair of linear equations given by Koza, there can be two or more legal combinations of these constants, symbols, and operators. Unless the types of problems to be processed are predetermined, it could be difficult to define the rules to measure fitness for problems under many different categories.

Fourth, applicant's invention displays reconstructed formula as well as associated properties and lets user to fix any problem by providing interfaces for user to assign new properties and modify existing properties without resorting to constructing a set of possible combinations and then finding the "best fitness" one.

Fifth, applicant's invention lets parser to extract the relations among components of an equations or a formula, to generate model file based on the extracted relations, and then to create source code file from model file without resorting to S-expression. As long as the relations among components are clear, many methods known in the art can be used to generate source code.

Sixth, applicant's invention provides user opportunity to modify source code at different levels. User can modify source code by modifying corresponding formula, modifying and assigning properties to the formula, modifying the generated model file, and modifying the source code file directly.

Seventh, applicant's invention provides interfaces for user to assign properties to a curve, select an interpolation method, specify the desired precision, and save results in database for reconstructing the curve in later use. Application lets user to decide that what is the best for a particular curve fitting without letting user play game to find the best fitness. Theoretically, if the "best fitness" is the concerned, for any curve with n points, a curve fitting based on a polynomial with degree of n-1 will guarantee a perfect fit on all n points.

Eighth, applicant's invention calls a corresponding library function directly to find the interpolation coefficients based on specified properties, interpolation method, and selected set of points without resorting to creating a population of S-expressions, finding the "best fitness" one, generating source code, and making symbolic regression.

Ninth, applicant's invention lets user to make decision on which methods will be used without resorting to a large set of non linear operations to solve a pair of linear equations.

Tenth, applicant's invention provides flow chart diagram for how to deal with formula block and curve block. Not only that Koza did not give any flow chart in any way similar to the ones in applicant's invention, but also Koza did not show any flow chart how to deal with formula block and curve block at all.

Pagallo wrote on column 9, lines 19 to 49:

> "FIG. 6 is used to illustrate the structure of a word unit 104 in greater detail. ... A word unit 104 includes a data portion 106 and a procedure portion 108. The data portion 106 includes a generic section 110, a subunit section 112, and an interpretation section 114. The generic section includes data types which will be found in all unit structures. The subunit section 112 will contain subunits of the unit 104. ... The interpretation data

section 114 is a list of the interpretations for the unit 104 with their rankings. ... The procedures portion 108 include methods and processes which allow the unit to operate as an object." (underlines added for emphasis)

Pagallo further showed on Fig. 6 what contained in subunits are the input strokes which labeled as UNITSTROKE1 "3", UNITSTROKE2 "3", UNITSTROKE3 "." with a reference number 112.

Pagallo also wrote on column 9 line 50 to column 10 line 9:

"FIG. 7 is used to illustrate the structure of an exemplary math symbol unit 116. ... Math symbol unit 116 includes a data portion 118 and a procedures portion 120. The data portion 118 includes a generic section 122, a subunit section 124, and an interpretation section 126. As before, the generic section will include the type [same type as the one for a word unit] .... The subunit section 124 will contain subunits of the unit 116. ... The interpretation data section 126 is a list of the interpretations for the unit 116 with their rankings. ... The procedures portion 120 include methods and processes which allow the unit to operate as an object." ([...] added for comment and underlines added for emphasis)

Pagallo further showed on Fig. 7 what contained in subunits are the all possible input strings labeled as "UNITWORD 33.2, UNITWORD 33,2 UNITWORD 37.2" with a reference number 124.

From Fig. 6 and Fig. 7 and corresponding descriptions, it is clearly that Pagallo has assigned fixed attributes to symbols with their associated values are allowed to change.

First, as Pagallo wrote, "The generic section includes data types which will be found in all unit structures", the generic section must contain fixed attributes only otherwise it would be impossible to "be found in all unit structures".

Second, as Pagallo wrote "The subunit section 112 will contain subunits of the unit 104" and as the examples shown by Pagallo, the subunit section contain only the possible data with predetermined data type and therefore the subunit section contain fixed attributes only. In word unit, the subunits are used to record the input strokes; in math symbol unit, the subunit are used to record all the possible different symbols so that "allow the unit to operate as an object". In other words, these subunits contain only the data to be processed.

Third, suppose that the subunit could have flexible attributes, especially newly defined attributes, but Pagallo had never mentioned how to evaluate the rank for symbols and equations with newly attributes. In fact, what Pagallo suggested worked for symbols and equations with predefined attributes because the values associated with the attributes could be used to determine corresponding ranks of symbols and equations. However, Pagallo's system could not work on a newly added attribute because Pagallo had not told his system how to handle the newly defined attributes.

Fourth, even though Pagallo can assign and change the values associated with predefined attributes, Pagallo never mentioned that his system allowed to define new attributes, assign the new attribute to related object, and change from one attribute to a different attribute. To assign and change a value associated with an attribute is a very different concept from to

assign and change an attribute to a different attribute. To assign and change a value associated with an attribute means that the general category defined by the attribute is still the same and different values specify different degree in the general category. To change an attribute to a different one or add a new attribute means to switch to a different category or define a new category.

Pagallo wrote on column 4 lines 3 to 8:

> "passing the stroke units to a number of recognition domains, where at least one of the domains is capable of parsing equations into a parsed tree utilizing a constrained attribute grammar; deriving a result of the equation (if requested) utilizing the parsed tree; and displaying the result on the screen proximate to the equation. The method also includes the step of receiving at least one stoke unit which comprises an edit to equation on screen; determining whether the edit changes the structures of the parsed tree; and editing the parsed tree if the edit does not changed its structure." (underlines added for emphasis)

On column 5 lines 5 to 6, Pagallo wrote:

> "An advantage of the present invention is that hand-written equations can be entered into a computer system, solved, and the results can be displayed on an output screen of the computer system."

On column 5 lines 7 to 9, Pagallo wrote:

> "Another advantage of the present invention is that, under some circumstances, a hand-written equation can be edited without re-parsing the entire equation."

From above paragraphs as well as the descriptions along with Fig. 6 and Fig. 7, it is very clear that

Pagallo did reconstruct equations and display the reconstructed. What displayed on screen are the equations from hand-written inputs or edited from hand-written as "under some circumstances, a hand-written equation can be edited" and the result derived from equation. These edited equations are different from the reconstructed equations disclosed by the applicant. As clearly stated by Pagallo, the edited equations are the hand-written equations modified by user. However, according to applicant's invention, the reconstructed equations are built based on the properties assigned to the components of these equations and the relations identified among these components, and further user will not directly involve in the process of reconstructing equations.

Second, suppose the displayed equations are reconstructed ones, then there would be no need for generating a set of possible forms for symbols, words, and equations and then evaluating their ranks.

Third, Pagallo did not mention that his system allows user to modify the properties associated with equations and symbols. As one can see from the recitation from Pagallo's patent, Pagallo allowed user to modify hand-written equations but not properties associated with the equations

Fourth, the attributes mentioned by Pagallo are predefined attributes, or fixed attributes.

Pagallo assigned some fixed attributes to words and Pagallo tried to identify the correct form of a hand-written word by generating a set of possible words, evaluating the ranks associated these possible words, and finding the best one. Pagallo never mentioned his system allowed user to define new attributes, assign new attributes, and modify new attributes.

Applicant takes different approaches to deal with equations

First, the system in the application emphasizes the relations among different components of an equation by providing interfaces for user to modify existing properties and assign new properties to related equations and its components to help identifying the relations. The system assumes the input image is in good condition, there is no any problem to recognize symbols in an equation or all available techniques in the art can correct any misidentifying problem. Pagallo emphasized to remove the errors associated with symbol identification.

Second, the system in the application provides interfaces for user to display reconstructed equations and lets user to examine various properties associated with the reconstructed equations and their components. Pagallo did not mention how to reconstruct equations, especially how to reconstruct equations based properties and identified relations, and display the reconstructed equations.

Third, the system in the application relies on the most intelligent resource – user to identify relation problem directly. By examining the reconstructed equations, user can notice relation problem. By examining and modifying associated properties, user can further specify how user wants to generate code. The purpose of applicant's system is to help user to verify results and user has all information needed for obtaining the result of an equation. Pagallo constructed a set of possible forms of words, symbols, and equations, found out the ranks with each item in the set, and then found the best one based on the ranks. In other words, Pagallo relied on the completeness of possible forms as well as the way to define rank. A purpose of Pagallo's system is to find the best equation candidate for a hand-written equation.

Fourth, the system in the application provides a mechanics for user to specify as few properties as possible. The system provides interfaces for user to set up default settings, one type of the default settings is to specify the default properties for commonly used symbols and functions for a subject, a subsection, and a particular instance. Then the system provides interfaces for user to modify properties and add new properties to give more information for guiding the system to generate code. Further, the system assigns properties to current block under processing based on the accumulated information to handle previous blocks. For example, suppose that one of the default properties associated with I0(x) links to Bessel Function of order 0 and one of the default properties associated with I1(x) links to Bessel Function of order 1. During processing one of previous blocks, if there is I0() in a previous block and user has modified its corresponding property to link to a different function, then the system will link the corresponding property of I1() to a similar one or ask user to specify a function. In addition, the system displays the reconstructed formula, provides interfaces for user to modify existing properties and add new properties to related symbols and functions. These gradually updated properties will guide the system on how to handle the formula, just like a professor will give his student enough hints to make sure the student can move forward but no more than necessary. Pagallo did not provide a mechanics to reduce gradually user's involvement.

Fifth, the system in the application assigns properties to equations and their components, finds the relation among these components, reconstructs equations, displays the reconstructed equations as well as associated properties, lets user gradually refine equations and their properties, and provides opportunity for user to modify source code at different levels. Pagallo did not provide such a system.

The Application Is Not Obvious Over Pagallo's Patent and Koza's Patent

Each years, thousands and thousands Ph. D. students, professors, and engineers spent tremendous time to reconstruct the program code needed to duplicate the results shown on journals, various proposals, and various standard materials for better understanding, verifying, and improving. Some software such as Matlab and Mathcad do help peoples to be able to repeat results sooner by providing some commonly used functions and symbolic calculation, but people still have to write the program by themselves, find out the parameters associated with curves, and then run under particular environment. This process will become more complex if the results are based on some experiential data and parameters. Writing code by hand does not only take time, but also is inconvenience. People wants to spend less time on duplicating results and spend more time on critical thinking, and not in reverse way.

If the application were obvious over Pagallo's patent and Koza's patent, software similar to what applicant disclosed should be available already. Applicant only find software Readiris, which converts image into text file but has not find any software which can generate programs directly from image, provide interfaces for user to assign and modify properties, reconstruct formulas, display the reconstructed formula, allows user to further modify related properties, let user to modify program, and display the results. Since society does not such a software, Koza's patent has been public available for about 13 years, and Pagallo's patent has been public available for about 9 years, there should be one of ordinary skill in the art to combine Pagallo's patent and Koza's patent together and there should be such software available already.

Koza and Pagallo Do Not Contain Any Justification To Support Their Combination, Much Less In The Manner Proposed

With Regard to the proposed combination of Koza and Pagallo, Pagallo did not refer any possible combination of his patent with Koza's patent, nor Pagallo mentioned Koza's patent, even these two patents are closely related and Pagallo's patent is about 5 years later than Koza's. Neither Pagallo nor Koza nor anyone suggested that there was a possible to combine two methods together not mention to give a clear picture as disclosed by the application. It is well known that in order for any prior-art references themselves to be validly combined for use in a prior-art §103 rejection, *the references themselves* (or some other prior art) must suggest that they be combined. E.g., as was stated in In re Sernaker, 217 U.S.P.Q. 1, 6 (C.A.F.C. 1993):

> "[P]rior art references in combination do not make an invention obvious unless something in the prior art references would suggest the advantage to be derived from combining their teachings."

That the suggestion to combine the references should not come form applicant was forcefully stated in Orthopedic Equipment Co. v. United States, 217 U.S.P.Q. 193, 199 (CAFC 1983):

> "It is wrong to use the patent in suit [here the patent application] as guide through the maze of prior art references, combining the right references in the right way to achieve the result of the claims in suit [here the claims pending]. Monday morning quarterbacking is quite improper when resolving the question of non-obviousness in a court of law [here the PTO]."

As was further stated in Uniroyal, Inc v. Rudkin-Wiley Corp., 5 U.S.P.Q. 2d 1434 (C.A.F.C. 1988), "[w]here prior-art reference require selective combination by the court to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gleaned from the invention itself ... *Something in the prior art must suggest the desirability and thus obviousness of making the combination.*" [Emphasis supplied.]

In line with these decisions, the Board stated in Ex parte Levengood, 28 U.S.P.Q. 2d 1300 (P.T.O.B.A. & I, 1993):

> "In order to establish a *prima facie* case of obviousness, it is necessary for the examiner to present *evidence*, preferably in the form of some teaching, suggestion, incentive or inference in the applied prior art, or in the form of generally available knowledge, that one having ordinary skill in the art *would have been led* to combine the relevant teachings of the applied references in the proposed manner to arrive at the claimed invention. ... That which is within the capabilities of one skilled in the art is not synonymous with obviousness. ... That one can *reconstruct* and/or explain the theoretical mechanism of an invention by means of logic and sound scientific reasoning does not afford the basis for an obviousness conclusion unless that logic and reasoning also supplied sufficient impetus to have led one of ordinary skill in the art to combine the teachings of references to make the claimed invention. ... Our reviewing courts have often advised the Patent and Trademark Office that it can satisfy the burden of establishing a *prima facie* case of obviousness only by showing some objective teaching in either the prior art, or knowledge generally available to one of ordinary skill in the art, that 'would lead' that individual 'to combine the relevant teachings of the reference.' ... Accordingly, an examiner cannot establish obviousness by locating references which describe various aspects of a patent applicant's invention without providing evidence of the motivating force which would impel one skilled in the art to do what the patent applicant has done."

In the present case, applicant has read the Koza's patent many times, especially column 15, lines 4-48 and column 55, lines 6-58 as indicated by examiner, applicant have not found any statement in Koza's patent that the information found is added to a database and the equations and curves will be processed in a way similar to what specified by applicant. In addition, there is no any description and explanation on how to combine Koza's patent and Pagallo's patent from Koza, Pagallo, or the Examiner. The fact that both reference teach people how to find the best fitness or rank of candidate equations is not an issue of applicant's invention and is not sufficient to gratuitously and selectively substitute parts of one reference for a part of another reference in order to meet applicant's novel combination.

The O.A. noted (p. 4) that the combination of Pagallo and Koza produces an advantage (increase the problem solving ability). Applicant submits that the fact that combination produces advantages militates in favor of applicant because it proves that the combination produces new and unexpected results and hence is unobvious.

As stated in the above Levengood case,

> "That one can *reconstruct* and/or explain the theoretical mechanism of an invention by means of logic and sound reasoning does not afford the basis for an obviousness conclusion unless that logic and reasoning also supplies sufficient impetus to have led one of ordinary skill in the art to combine the teachings of the references to make the claimed invention."

Applicant therefore submits that combining Pagallo and Koza is not legally justified and therefore is improper and thus applicant submits that the rejection on there references is also improper and should be withdrawn.

Applicant respectively request, if the claims are again rejected upon any combination of references, that the Examiner recite the sentences from references and include an explanation, in accordance with M.P.E.P. § 706.02, Ex parte Clapp, 27 U.S.P.Q. 972 (P.O.B.A. 1985), and Ex parte Levengood, supra, "a factual basis to support his conclusion that it would have been obvious" to make the combination.

One Skilled In the Art Would Have No Reason to Make Combination of Pagallo's Patent and Koza's Patent

The combination could not be made in a useful way because Pagallo has his own way to parse equation by building a set of different possible equations and finding the best one and does not need to build another a set of S-expression. Pagallo built a set of possible candidate words, candidate symbols, and candidate equations, defined the rule to find ranks, found the ranks, located the best candidate, and finally parsed the best candidate. Koza built a set of candidate S-expressions, defined "fitness", found the best S-expression, and then generated code based on the best S-expression. Essentially these two patents did the same things and used different approaches and there is no need for combining them together. It will be ridiculous for one to use Pagallo's method to find the best candidate and then to use Koza's method to find best S-expression and then parse the S-expression. In other words, these two patents took different approaches for solving a similar problem. This might be one of the possible reasons Pagallo did not refer Koza's patent and not even mentioned Koza's patent at all.

Even If Pagallo and Koza Were to Be Combined In The Manner Proposed, The Proposed Combination Would Not Show All of The Novel Features of Claim 21.

However even if the combination of Pagallo and Koza were legally justified, claim 21 would still have novel and unobvious features over the proposed combination. In other words, applicant's invention, as defined by claim 21, comprises much more than merely substituting a plurality of templates for one template.

Specifically, clauses of claim 21 listed below clearly distinguish applicant's approach from Pagallo and Koza's, or any possible combination thereof, since these clauses recite:

"means for selecting portions from said information source, said portions comprising formulas and figures;

...

means for creating identifications to distinguish among said items;

...

means for modifying said properties to change said desired relations among said items and said desired ways of handling said block and said items;

means for defining new properties and means for assigning said new properties to said block and said items to provide specific requirements on said block and said items;

...

means for displaying said block and said items;

means for displaying properties associated with said block and said items;

means for adjusting properties to be assigned to said block and said items by using information obtained from handling previous blocks".

Thus applicant submit that his invention is much more than merely substituting a plurality of templates for one template and that claim 21 clearly recites novel subject matter which distinguishes over any possible combination of Pagallo and Koza.

The Novel Features of Claim 21 Produce New And Unexpected Results And Hence Are Unobvious And Patentable Over These References Under § 103.

Also applicant submit that novel features of claim 21 are also unobvious and hence patentable under §103 since they produce new and unexpected results over Pagallo and Koza, or any combination thereof.

These new and unexpected results are the ability of applicant's system to define new properties, modify properties, display reconstructed equations and formula as well as associated properties, learn from processing previous formulas and equations, and modify code at different levels. An additional major new and unexpected result is that the applicant's system will not fail to provide a result for a formula or an equation because the functions in the formula or the equation are either in database already, or defined on papers where the formula or the equation locates, or commonly used functions that user should know how to build them. In order words, user does not need to collect a set of candidate equations or formulas and define rank and fitness. The novel features of applicant's system which effect these differences are, as stated, clearly recited in claim 21.

The Dependent Claims 22-34 Are A Fortiori Patentable Over Pagallo and Koza
New dependent claims 22-34 incorporate all the subject matter of claim 21 and add additional subject matter, which makes them a fortiori and independently patentable over these references.

Claim 22 additionally recites:
"means for generating model file and means for modifying said model file, wherein said model file is a file that contains structure information of said source code file."

This is entirely foreign to Pagallo and Koza, or any combination thereof since, as stated, the systems of these references do not provide opportunity to user to modify source cod file indirectly by modifying the intermediate model file.

Claim 23 adds "associating default properties to said items and said block" and "modifying said default properties".

Claim 24 recites "adjusting and assigning properties to said previous blocks by making use of information accumulated for handling said block and said items that are under processing". Again this is clearly foreign to Pagallo and Koza.

Claim 25 adds "displaying and hiding selected aspects of said block and said items"

Claims 26 and 27 specify more details for processing curve. Pagallo and Koza did not provide a scheme similar to what disclosed by Claims 26 and 27.

Claims 28 to 30 specify more details for processing formula. Pagallo and Koza did not provide a scheme similar to what disclosed by Claims 28 to 30.

Claim 31 recites more details for "displaying properties". Neither Pagallo nor Koza nor their combination disclosed any scheme similar to what described by claim 31.

Claim 32 recites more details for "generating tokens by a scanner".

Claim 33 recites more details for "generating symbol tables by a parser".

Claim 34 recites more details for "updating database".

The Independent Claims 35 and 38 Are Written Based On the Suggestion from the Examiner in Our Telephone Conversation
Based on Fig. 5 of the application, the applicant writes the independent claim 35 to specify the major operations needed for processing formula. Based on Fig. 4 of the application, the applicant writes the independent claim 38 to specify major operations needed for processing curve. Since both two figures are foreign to either Pagallo or Koza or their combination and the

claims clearly define the novelties and features, both these independent claim are in condition for allowance.

The Dependent Claims 36, 37, 39 and 40
Claim 36 states that user can modify properties associated with related formula and define new properties.

Claim 37 specifies more detail for "processing formula block".

Claim 39 states more details for "extracting information from said curve".

Claim 40 describes various methods "selecting one curve from a group of separated curves".

Conclusion
For all of above reasons, the applicant submits that claims are now in proper from, and that the claims all define patentably over the prior art. Therefore the applicant submits that this application is now in condition for allowance, which action the applicant respectfully solicits.

Conditional Request for Constructive Assistance
The applicant has amended the claims of this application so that they are proper, definite, and related, and define novel features, which are also unobvious. If, for any reason this application is not believed to be in full condition for allowance, the applicant respectfully requests the constructive assistance and suggestions of the Examiner pursuant to M.P.E.P § 2173.02 and § 707.07(j) in order that the undersigned can place this application in allowable condition as soon as possible and without the need for further proceedings.

Very respectfully,

George L. Yang

---------- Applicant Pro Se ----------

4628 Kings Mill Way,
Owings Mills, MD 21117
Tel: (410)581-0969
Email: george_yang@yahoo.com